IN THE CLAIMS:

1   1. (Original) A method for converting a file access data structure from a first endianness

2   to a second endianness, the method comprising the steps of:

3   identifying, from a descriptor look up table, a series of actions to perform on ele-

4   ments of the file access data structure; and

5   performing the identified series of actions on the elements of the file access data

6   structure.

1   2. (Original) A method of converting elements of a file access data structure from a first

2   endianness to a second endianness, the method comprising the steps of:

3   determining if the file access data structure is a critical path data structure;

4   converting, in response to the file access data structure being a critical path data

5   structure, the elements from the first endianness to the second endianness using a set of

6   specific code functions;

7   converting, in response to the file access data structure not being a critical path

8   data structure, a header of the file access data structure from the first endianness to the

9   second endianness using a second set of specific code functions; and

10   calling a byte swapping engine to convert selected elements of the file access data

11   structure from the first byte order to the second byte order.

4

1 3. (Original) The method of claim 2 wherein the file access data structure further com-

2 prises a direct access file access data structure.

1 4. (Original) A system for converting elements of a file access data structure from a first

2 endianness to a second endianness, the system comprising:

3 an input buffer, the input buffer storing the file access data structure to be con-

4 verted;

5 a byte swapping engine, the byte swapping engine operative interconnected with a

6 descriptor table; and

7 an output buffer, the byte swapping engine placing the file access data structure in

8 the output buffer after conversion.

1 5. (Original) The system of claim 4 wherein the descriptor table further comprises a set

2 of entries describing various file access data structures, each entry further comprising a

3 size field and an operation field.

1 6. (Original) The system of claim 4 wherein the file access data structure further com-

2 prises a direct access file access data structure.

1 7. (Original) A method for converting a data structure from a first byte order to a second

2 byte order, the method comprising the steps of:

3 reading an element entry from a descriptor table;

5

4   performing an action on an element of the data structure, the action being defined

5   in the element entry read from the descriptor table; and

6   placing the element in an output buffer.

1   8. (Original) The method of claim 7 wherein the step of performing an action on an ele-

2   ment further comprises the step of copying the element from an input buffer to the output

3   buffer.

1   9. (Original) The method of claim 7 wherein the step of performing an action on an ele-

2   ment further comprises the step of byte swapping the element.

1   10. (Original) The method of claim 7 wherein the element entry of the descriptor table

2   further comprises a field describing a size of the element and a field describing an action

3   to be performed.

1   11. (Original) A file server for use in a network storage environment, the file server

2   comprising:

3   a byte swapping engine, the byte swapping engine performing a defined operation

4   on each of a plurality of elements of a file access data structure.

1   12. (Original) The file server of claim 11 wherein the file server further comprises a de-

2   scriptor look up table, the descriptor look up table having a plurality of entries, each of

3   the plurality of entries associated with a specific file access data structure.

1   13. (Original) The file server of claim 12 wherein each of the plurality of entries further

2   comprises a plurality of elements, each of the elements having a size field and an opera-

3   tion field.

1   14. (Original) The file server of claim 13 wherein the defined operation is defined by the

2   operation field of the entry associated with the file access data structure.

1   15. (Original) A computer-readable medium, including program instructions executing

2   on a computer, for converting elements of a file access data structure from a first endian-

3   ness to a second endianness, the method comprising the steps of:

4   determining if the file access data structure is a critical path data structure;

5   converting, in response to the file access data structure being a critical path data

6   structure, the elements from the first endianness to the second endianness using a set of

7   specific code functions;

8   converting, in response to the file access data structure not being a critical path

9   data structure, a header of the file access data structure from the first endianness to the

10   second endianness using a second set of specific code functions; and

11   calling a byte swapping engine to convert selected elements of the file access data

12   structure from the first byte order to the second byte order.

1   16. (Original) A method for converting elements of a file access data structure from a

2   first endianness to a second endianness, the method comprising the steps of:

3   determining a type of the file access data structure;

4   processing, in response to the file access data structure of being of a first type, the

5   file access data structure along a first processing path;

6   processing, in response to the file access data structure being of a second type, the

7   file access data structure along a second processing path.

1   17. (Original) The method of claim 16 wherein the first type further comprises a critical

2   path data structure.

1   18. (Original) The method of claim 16 wherein the first processing path further com-

2   prises a set of specifically coded functions.

1   19. (Original) The method of claim 16 wherein the second processing path further com-

2   prises a byte swapping engine.

Please add new claims 20, et seq., as follows:

1 20. (New) A method for converting a data structure, comprising:

2   calling a byte-swapping engine;

3   providing a file access data structure as input to the byte-swapping engine;

4   providing a descriptor look up table to the byte-swapping engine;

5   identifying, from the descriptor look up table, a series of actions to perform on

6 elements of the file access data structure in order to swap bytes of the file access data

7 structure from a first endianness to a second endianness; and

8   performing the identified series of actions on the elements of the file access data

9 structure.

1 21. (New) The method as in claim 20, further comprising:

2   using as the file access data structure a file having Direct Access File System

3 (DAFS) protocol.

1 22. (New) The method as in claim 20, further comprising:

2   determining if the file access data structure is a critical path data structure, and if

3 it is, perform byte swap operations using specific code functions.

1   23. (New) The method as in claim 20, further comprising:

2   determining if the file access data structure is a critical path data structure, and if

3   it is not, perform byte swap operations on a data structure header.

1   24. (New) The method as in claim 20, further comprising:

2   swapping bytes of the data structure as needed, in response to swapping bytes of

3   the file access data structure.

1   25. (New) The method as in claim 20, further comprising:

2   determining if an element entry of the descriptor look up table is nested;

3   branching to the nested entry;

4   identifying, from the descriptor look up table, a series of actions to perform on

5   elements of the nested entry in order to swap bytes of the entry from a first endianness to

6   a second endianness.

1   26. (New) A computer to convert a data structure, comprising:

2   means for calling a byte-swapping engine;

3   means for providing a file access data structure as input to the byte-swapping en-

4   gine;

5   means for providing a descriptor look up table to the byte-swapping engine;

6 means for identifying, from the descriptor look up table, a series of actions to per-

7 form on elements of the file access data structure in order to swap bytes of the file access

8 data structure from a first endianness to a second endianness; and

9 means for performing the identified series of actions on the elements of the file

10 access data structure.

1 27. (New) The computer as in claim 26, further comprising:

2 means for using as the file access data structure a file having Direct Access File

3 System (DAFS) protocol.

1 28. (New) The computer as in claim 26, further comprising:

2 means for determining if the file access data structure is a critical path data struc-

3 ture, and if it is, perform byte swap operations using specific code functions.

1 29. (New) The computer as in claim 26, further comprising:

2 means for determining if the file access data structure is a critical path data struc-

3 ture, and if it is not, perform byte swap operations on a data structure header.

1 30. (New) The computer as in claim 26, further comprising:

2 means for swapping bytes of the data structure as needed, in response to swapping

3 bytes of the file access data structure.

1   31. (New) The computer as in claim 26, further comprising:

2   means for determining if an element entry of the descriptor look up table is

3   nested;

4   means for branching to the nested entry;

5   means for identifying, from the descriptor look up table, a series of actions to per-

6   form on elements of the nested entry in order to swap bytes of the entry from a first en-

7   dianness to a second endianness.

1   32. (New) A computer readable media, comprising:

2   said computer readable media containing instructions for execution on a processor

3   for the practice of a method for converting a data structure, the method having the steps

4   of,

5   calling a byte-swapping engine;

6   providing a file access data structure as input to the byte-swapping engine;

7   providing a descriptor look up table to the byte-swapping engine;

8   identifying, from the descriptor look up table, a series of actions to perform on

9   elements of the file access data structure in order to swap bytes of the file access data

10   structure from a first endianness to a second endianness; and

11   performing the identified series of actions on the elements of the file access data

12   structure.

1   33. (New) Electromagnetic signals propagating on a computer network, comprising:

2   said electromagnetic signals carrying instructions for execution on a processor for

3   the practice of a method for converting a data structure, the method having the steps of,

4   calling a byte-swapping engine;

5   providing a file access data structure as input to the byte-swapping engine;

6   providing a descriptor look up table to the byte-swapping engine;

7   identifying, from the descriptor look up table, a series of actions to perform on

8   elements of the file access data structure in order to swap bytes of the file access data

9   structure from a first endianness to a second endianness; and

10   performing the identified series of actions on the elements of the file access data

11   structure.